البرمجة المقترحة:
لغة C++ . ② - تعقيد الخوارزمية . ③ - العلاقات العودية .
④ - خوارزميات الترتيب . ⑤ - خوارزميات البحث .

مفردات اللغة (C++) :
1- الأحرف الكبيرة : A, B, ... , Z
2- الأحرف الصغيرة : a, b, ... , z
3- الأرقام : 0, 1, 2, ... , 9
4- الرموز والإشارات : + - > < = ! ...

الكلمات المحجوزة : هناك بعض الكلمات القياسية المحجوزة من قبل مترجم لغة C++ ومن هذه الكلمات :
include , int , Long , const , float , double , ...
وهذه الكلمات تكتب بأحرف صغيرة .

الأسماء التعريفية : ونقصد بها أسماء المتغيرات وأسماء الثوابت وأسماء البرامج الفرعية وأسماء السجلات .
عند كتابة الأسم التعريفي أن يبدأ بحرف ويجب أن لا يحوي الإسم على فراغات ورموز أو إشارات ويتضمن الأسم أحرف وأرقام .
لغة C++ تفرق بين الأحرف الكبيرة والأحرف الصغيرة لذلك يمكن التعامل بين الكلمات التالية :
PROGRAM , Program , PROGram
والأسم لا يتضمن الكلمات المحجوزة .

عملية الإسناد (=) :
عند إسناد قيمة لمتغير نضع في الطرف الأيسر أسم المتغير وفي الطرف الأيمن القيمة ونفصلها :
(; القيمة = اسم المتغير)
القيمة الموجودة في الطرف الأيمن يمكن أن تأخذ عدة حالات :
(قيمة عددية - اسم متغير يملك قيمة - تعبير حسابي للدالة الإضافية)
قيمة عددية : A = 5 ;
اسم متغير يملك قيمة : B = A ;
تعبير حسابي : C : 2*A + 8

m.t

Generated by CamScanner from intsig.com

Subject .... Date ....

- أنواع المتغيرات :

أثناء كتابة البرنامج لابد من تعريف جميع المتغيرات المستخدمة مع تحديد هذه المتغيرات وأنواع متغيرات بسيطة تأخذ قيمة واحدة فقط وهناك متغيرات المركبة تأخذ أكثر من قيمة وهناك متغيرات ترتيبية أي عند معرفة قيمته يمكن معرفة القيمة السابقة والقيمة التالية .

ومن هذه المتغيرات الترتيبية المتغيرات الصحيحة، ومن المتغيرات غير الترتيبية المتغيرات الكسرية .

والمتغيرات المركبة تأخذ مثل سجلات ومصفوفات .

- نوع المتغيرات الصحيحة :

وهو أحد المتغيرات الترتيبية وتتضمن القيم السالبة والصفر والقيم الموجبة .

- هناك عدة أنواع للمتغيرات الصحيحة كل منها يختلف عن الآخر حسب الحجم في الذاكرة وأيضاً حسب مجال قيمتها (القيمة) وهي موضحة بالجدول الآتي :

نوع المتغيرات	الحجم في الذاكرة	مجال القيم
int	2 byte	[-32868 , +32867]
Long	4 byte	-214 , 748 , 364 , 8 →

- العمليات الحسابية على المتغيرات الصحيحة :

(+)   (−)   (*)   (/) → (قسمة عددية صحيحة)

8/2 → 4 , 7/2 → 3 , 2/5 → 0

عملية الزيادة : وهي عملية زيادة المتغيرة بمقدار واحد. ++X

" النقصان : " " نقصان " " " " . --X

أي أن : ++X ⇔ X = X+1

--X ⇔ X = X-1

- الفرق بين (++X) و (X++) :

إن (++X) و (X++) كلاهما تعني زيادة قيمة (X) بمقدار واحد إلا أنه عند استخدامها في التعابير الحسابية يوجد فرق بينهما .

- عند استخدام (++X) في التعبير الحسابي يتم أولاً زيادة قيمة (X) بمقدار واحد ثم بعد ذلك تحسب قيمة التعبير الحسابي .

Generated by CamScanner from intsig.com

Subject
Date

- أما عند استخدام (X++) فيتم أولاً حساب قيمة التعبير ثم بعد ذلك يتم زيادة قيمة المتغير (X) بمقدار واحد.

مثال:

```
int x , y , z ;
x = 0 ;      y = 0 ;   → بداية الأمر
z = ++x + ++y ; 2
     1     1
z = ++x + y++ ; 3
     2    2 → بعد حساب القيمة
z = x++ + ++y ; 5
     3 ← بعد حساب القيمة    3
z = ++x + ++y ; 8
```

(نحسب قيمة التعبير أولاً وبعد ثم نزيد قيمة المتغير)

- **إيعازات المعالج:**

عند كتابة البرنامج بلغة C++ نبدأ البرنامج بأحد إيعازات المعالج (أمر توجيه).

1- أمر التوجيه (include): عند كتابة البرنامج نبدأ البرنامج بأمر التوجيه (include).

```
# include < اسم المكتبة >
```
لا يوجد فارق

- يوجد العديد من المكتبات بلغة الـ C++ وكل مكتبة تختص ببعض الأوامر وأشهر هذه المكتبات:

```
# include <iostream.h>
```
input / output / تيار / header رأس

وهذه المكتبات تتعلق بالإدخال والإخراج.

```
# include <math.h>
```
تتعلق هذه المكتبات بالدوال الرياضية.

```
# include <string.h>
```
تتعلق هذه المكتبات بالسلاسل الحرفية.

```
# include <ctype.h>
```
وتتعلق هذه المكتبات بالأحرف.

```
# define  اسم الثابت  القيمة  ✗
# define  PI   3.14
# define  f    2.7
# define  n    3
```

AL AMANY Notebooks

Generated by CamScanner from intsig.com

Subject "المحاضرة الثانية" Date

m.t

- ملاحظات على عملية الإسناد

(1)

```
int x , y , z ;
x = 0 ; y = 0 ; z = 0 ;
x = y = z = 0 ;
```

(2)

```
x = x+2 ;   ⇔   x+ = 2 ;
x = x-2 ;   ⇔   x- = 2 ;
x = x*2 ;   ⇔   x* = 2 ;
```

x+ = 2 → تقرأ x تساوي x+2

(3)

```
y = 2*x+5 ;
z = y ;       ⇒   z = y = 2*x+5 ;
```

فنستطيع أن نستخدم أكثر من أمر.

* المتغيرات الكسرية : يوجد لدينا أنواع للمتغيرات الكسرية كل منها يختلف عن الآخر حسب الحجم من الذاكرة وحسب مجال القيم.

النوع	الحجم من الذاكرة	مجال القيم
float	4byte	$-3.4\times10^{-38} \to +3.4\times10^{+38}$
double	8 byte	$-1.7\times10^{-308} \to 1.7\times10^{+308}$

- إيعازات المعالج

- الدالة abs() : تستخدم هذه الدالة من أجل حساب القيمة المطلقة للعدد الصحيح والصغير.

```
y = abs(x);
```

- الدالة Fabs() : تستخدم هذه الدالة لحساب القيمة المطلقة للعدد الكسري.

```
y = fabs(x);
```

عدد كسري

- الدالة sqrt() : تستخدم لحساب الجذر التربيعي للعدد.

```
y = sqrt(x);
```

- الدالة exp() : الدالة الأسية.

```
y = exp(x);
```

- الدالة Log() : الدالة اللوغاريتمية

```
y = Log(x);
```

AL AMANY Notebooks

Generated by CamScanner from intsig.com

Subject | Date |

- الدالة Pow:

$y = pow(a, b);\ a^b$

- الدالة sin(x) و cos(x): x تقدر بالزوايا ولا تقدر بالدرجة.

$$\begin{matrix} 180 & \pi \\ \text{الزاوية} & x \end{matrix} \Rightarrow x = \frac{\text{الزاوية المعطاة} \times \pi}{180}$$

- الدالة Floor( ): تستخدم هذه الدالة من أجل تحويل القيم الكسرية الى قيم صحيحة

اذا كان العدد الكسري سالب يهمل الجزء الكسري وينقص بمقدار واحد.

واذا كان الجزء الكسري موجب يهمل الجزء الكسري فقط.

مثال:

$x_1 = -5.7;\quad x_2 = 7.9;$

$y_1 = Floor(x_1); = -6$

$y_2 = Floor(x_2); = 7$

- الدالة Ceil( ): تستخدم هذه الدالة من أجل تحويل العدد الكسري الى عدد صحيح

اذا كان العدد الكسري سالب يهمل الجزء الكسري، واذا كان العدد كسري موجب نهمل العدد الكسري ونضيف واحد.

مثال:

$x_1 = -5.7;\quad x_2 = 7.9;$

$y_1 = Ceil(x_1); = -5$

$y_2 = Ceil(x_2); = 8$

[2] أمر الطباعة cout:

ويقرأ c out put (الإخراج للغة c) ويستخدم هذا الأمر من أجل طباعة نصية على الشاشة أو متغيرات:

```
cout << " العبارة النصية المراد طباعتها " ;
```

حتى نستخدم أمر الطباعة نعرف الملف الرأسي:

```
# include <iostream.h>
```

(مجرى الاتجاه الطباعة)

- مثال:

```
cout << "The first Program" ;
```

نتيجة التنفيذ: يطبع عبارة The first Program على الشاشة.

- يمكن أن نستخدم أمر الطباعة cout من أجل طباعة قيمة متغير

AL AMANY Notebooks

Generated by CamScanner from intsig.com

Subject ___ Date ___

مثال:

```
int X ;
X = 15 ;
Cout << X ;
```

- طباعة عبارة نصية وقيمة متغير.

```
Cout << "X=" << X ;
```

(عبارة) (قيمة)

مثال:

```
int X , y , z ;
X = 10 ; y = 20 ; z = 30 ;
Cout << X << y << z ;
```

ناتج الطباعة: 10 20 30 تتم على نفس السطر.

ملاحظة: مع الأمر Cout وضمن اشارات التنصيص المزدوجة يمكن أن نستخدم بعض الرموز التي تلي إشارة (\) ولكل رمز عمل محدد.

- \n : من أجل الانتقال الى سطر جديد ويوضع الرمز إما في بداية العبارة النصية أو في الوسط أو في النهاية

مثال:

```
Cout << "\n The first Program" ;
```

(الانتقال الى سطر جديد ثم طباعة العبارة)

```
cout << " the first Program \n" ;
```

(الطباعة العبارة ثم الانتقال الى سطر جديد)

- \n \n : انتقال الى سطرين.

- \t : القفز بمقدار ثمانية فراغات.
- \b : حذف الحرف الذي يقع على اليسار

مثال: مع المثال السابق

```
Cout << "\n X = " << X ;
Cout << "\n y = " << y ;
```

ناتج الطباعة:

X = 10
y = 20

AL AMANY Notebooks

Subject | Date |

3- أمر الإدخال cin : ويقرأ (c-in) ويستخدم هذا الأمر من أجل إدخال المتغيرات من لوحة المفاتيح وصيغته :

cin >> اسم المتغير

(أداة الإدخال)

ويستخدم ملف الرأسي <iostream.h>

مثال :-

```
int x ;
cin >> x ;
```

(حجز مكان أو موقع في الذاكرة حجمه 2 بايت واسمه X).

لحفظه من رقم البايت في الذاكرة .

m.t

* أولويات العمليات الحسابية :

24 / 6 * 2 → 8 * 2

1. الإشارة السالبة (-) ،

* / %

\+ -

- ملاحظة ① عند وجود معامل بين عمليتين لهما نفس درجة الأولوية ننفذ العملية التي تقع على اليسار .

② عند وجود معامل بين عمليتين لهما درجتي أولوية مختلفتين ننفذ العملية الأعلى .

③ عند وجود الأقواس ، ننفذ العملية التي بين قوسين .

* العمليات المنطقية : وهي :

> >= < <=

== ← مساواة منطقية .

!= (or / and / && / not)

(a > 2 && b < 5) مثال

&& (and) || (OR) ! (not)

AL AMANY Notebooks

Generated by CamScanner from intsig.com

حسابية ... وردان
عن حسابية ... وجزاف

Subject Date

x	y	x && y	x || y
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

تمرين ٢ - أكتب برنامج يجمع ويطرح ويضرب عدد بين صحيحين ؟

```
# include <iostream.h>        → مبدأ بالمكتبة
  main(  )                     → كل برنامج يبدأ بالدالة الرئيسية
{
   int x , y , z1 , z2 , z3 ;      x=10
cout << "\n x = " ; cin >> x ;     y=20
 cout << "\n y = " ; cin >> y ;
z1 = x + y ; cout << "\n z1 = " << z1 ;
z2 = x - y ; cout << "\n z2 = " << z2 ;
z3 = x * y ; cout << "\n z3 = " << z3 ;

return ( 0 ) ;  →
```

بوضع الأقواس داخلها المتغير
اذا لم توجد متغيرات تترك الأقواس فارغة

Void - كلمة محجوزة تستخدم مع التعامل مع الدالة على أنها اجراء
واذا وضعت void لا حاجة لـ return

← إعادة القيمة للصفر

Void - هي كلمة محجوزة وهو نوع متغيرات يعبر عنه بلا شيء ويستخدم مع الإجراءات مع لغة الـ C ، ويستخدم أيضاً مع المؤشرات.

m.t

AL AMANY Notebooks